

- Las respuestas deben estar escritas con bolígrafo negro o azul

Pregunta 1 (1 pto). Explica cómo se realiza el encaminamiento en una red hipercubo.

Pregunta 2 (1 pto). Explica la diferencia entre un protocolo de coherencia de memoria de invalidación y un protocolo de coherencia de memoria de actualización.

Pregunta 3 (1 pto). Dibuja una red delta con 9 entradas y 16 salidas utilizando conmutadores crossbar 3x4.

Pregunta 4 (2 ptos). Un banco ha adquirido un supercomputador formado por 32768 nodos conectados mediante una red toro 3D cuyos enlaces tienen una velocidad de 2Gbit/s. Para terminar de analizar el rendimiento del supercomputador se desea saber cuánto tardará un paquete formado por 56 bytes (incluyendo la cabecera) que se envía desde el nodo 3056 al nodo 12018. El tiempo de enrutamiento es de 9ns. Calcula los tiempos de envío tanto utilizando "store and forward" como "wormhole". Nota: la cabecera del paquete está formada por 2 bytes.

Pregunta 5 (1 pto) Explica brevemente qué tipos de paralelismo existen y en qué niveles puede aplicarse

Pregunta 6 (1 pto) 1. Explica en qué consiste el encadenamiento de operaciones en las máquinas vectoriales

Pregunta 7 (1 pto). Explica la diferencia entre predicción dinámica explícita y predicción dinámica implícita.

Pregunta 8 (2 puntos). Suponer un computador superescalar que dispone un buffer de reorden, que permite resolver los riesgos WAR y WAW, y una cola y ventana de instrucciones con un número de entradas suficiente. El procesador es capaz de captar 3 instrucciones por ciclo, decodificar, emitir y completar 2 instrucciones por ciclo. Además, la emisión y finalización de las instrucciones puede ser desordenada. Para las tareas de ejecución, se dispone de las siguientes unidades segmentadas: 2 FP mul/div (5c), 2 FP add (2c), 2 ALU int (1) y 2 load/store (3). Finalmente, se dispone de un predictor de saltos dinámico que utiliza BTB de 3 entradas y 2 bits de predicción. Cuando se añade una nueva entrada en el BTB, su primera predicción sería de estado A (salto efectivo) si el salto es hacia atrás y de estado D (salto no efectivo) si el salto es hacia adelante.

En el computador se ejecuta el siguiente fragmento de programa:

```
0x01      addi r3,r0,#4 ; r3=4
0x02      add r4,r0,r3 ; r4=r3
0x03  loop: subi r3,r3,#1 ; r3=r3-1
0x04      beqz r3,end; si r3=0 saltar end
0x05      ld f1,-8(r1) ; cargar a[i-1]
0x06      ld f2,0(r1) ; cargar a[i]
0x07      ld f3,0(r2); cargar b[i]
0x08      muld f4,f1,f2 ; a[i-1]*a[i]
```

```

0x09      muld f4,f4,f3 ; a[i-1]* a[i] * b[i]
0x0A      sd 0(r1),f4 ; almacenar a[i]
0x0B      addi r1,r1,#8; r1=r1+8
0x0C      addi r2,r2,#8; r2=r2+8
0x0D      bnez r4,loop
0x0E  end:  addd f4, f4, f4

```

- a) (1,5 puntos) Planificar las instrucciones utilizando una tabla como la siguiente hasta la primera iteración del bucle (sin realizar el salto). Suponer que inicialmente $r1=0$ y $r2=100$

inst	IF	ID/ISS	EX	ROB	WB	Comentario

- b) (0,25 puntos) Realizar una traza de ejecución del código, mostrando el contenido de la BTB, (BTB inicialmente vacía) para todas las iteraciones del bucle.

Dir salto	Dir destino	Bits predicción

- c) (0,25 puntos) Determinar el número de ciclos que tardaría en ejecutarse el código suponiendo que un fallo en la predicción del salto implica una penalización de 3 ciclos